

USING SOCIAL MEDIA AND MOBILE DEVICES TO DISCOVER AND SHARE DISASTER DATA PRODUCTS DERIVED FROM SATELLITES

Daniel Mandl¹, Patrice Cappelaere², Stuart Frye³, John Evans⁴, Karen Moe¹

¹ NASA/GSFC Code 581 Greenbelt, Md 20771 USA daniel.j.mandl@nasa.gov *

² Vightel Inc. Ellicott City, Md 21042 USA

³ SGT Inc. NASA/GSFC Greenbelt, Md 20770 USA

⁴ GST, Greenbelt, Md 20770 USA

⁵ Code 407 NASA/GSFC Greenbelt, Md 20771 USA

*Corresponding author, email: daniel.j.mandl@nasa.gov

Abstract - Data products derived from Earth observing satellites are difficult to find and share without specialized software and often times a highly paid and specialized staff. For our research effort, we endeavored to prototype a distributed architecture that depends on a standardized communication protocol and applications program interface (API) that makes it easy for anyone to discover and access disaster related data. Providers can easily supply the public with their disaster related products by building an adapter for our API. Users can use the API to browse and find products that relate to the disaster at hand, without a centralized catalogue, for example floods, and then are able to share that data via social media. Furthermore, a longer-term goal for this architecture is to enable other users who see the shared disaster product to be able to generate the same product for other areas of interest via simple point and click actions on the API on their mobile device. Furthermore, the user will be able to edit the data with on the ground local observations and return the updated information to the original repository of this information if configured for this function. This architecture leverages SensorWeb functionality [1] presented at previous IGARSS conferences.

The architecture is divided into two pieces, the frontend, which is the GeoSocial API, and the backend, which is a standardized disaster node that knows how to talk to other disaster nodes, and also can communicate with the GeoSocial API. The GeoSocial API, along with the disaster node basic functionality enables crowdsourcing and thus can leverage insitu observations by people external to a group to perform tasks such as improving water reference maps, which are maps of existing water before floods. This can lower the cost of generating precision water maps.

Keywords—Data Discovery, Disaster Decision Support, Disaster Management, Interoperability, CEOS WGISS Disaster Architecture

I. KEY FUNCTIONALITY

The present GeoSocial API is being prototyped. The functionality described in the paper provides the present state of the GeoSocial API. However, the components being used for the GeoSocial API evolve as various user groups use the API. The GeoSocial API depends on standardized interfaces for retrieving disaster data products. So as the various disaster nodes become available, some of the components and standards may change. The GeoSocial API builds on the underlying SensorWeb components that have been developed for disaster response and an effort to develop an interoperable disaster architecture.[1][2][3][4][5]

II. KEY FUNCTIONALITY

The key functionality that is described is for both the front end and backend components of the GeoSocial API. Beginning with the front end, the user interfaces with the API using a goal driven statement. “I want [verb][object][target]”. For example, “I want a current water extent of Haiti”. User agents query available disaster nodes and return behaviors (cross domain scripts) which can return the needed product. The user decides which of the behaviors to select and the products are then generated dynamically and returned to user. Activity streams are also pushed to followers of that user on social networks. Users see the product and mimic the behavior with other areas. The front end must be compatible with geojson standards.

The back end disaster nodes feed the GeoSocial API and has to have some key functionality as follows:

- Visualizer (Mapbox)
- Editor
- Operational database
- Change set database
- Social media interface

Note that the editor , operational database and change set database features have not been extensively exercised at this time. These capabilities will be further integrated when more experiments with crowdsourcing are performed in the next couple of years.



Fig 1 Basic overview of GeoSocial API/Disaster Node Functional Activity Flow

Figure 1 depicts the basic operations concept for the GeoSocial API/Disaster Node system. Note there are two methods to access disaster data products, the web based GeoSocial API or the mobile device App version of the GeoSocial API. Standardized queries return available products from disaster nodes for an area of interest. The user can retrieve one of the displayed products, share the products over social media and as a future capability be able to mimic the product for a different area of interest.



Fig 2 Sample Twitter display for Radarsat flood map

Figure 2 depicts a typical notification of an available flood extent product from Radarsat. The thumbnail picture shows a low resolution picture of the product. The user can click on the hyperlink and view the full resolution product.

Figure 3 depicts another scenario in which a product from the Autonomous Modular Sensor (AMS) which detects wild fires on the ground, is downlinked to the ground and stored in a disaster node. Subscribers to this type of data product are notified as shown via a tweet and then can view the full products using Github and Mapbox.

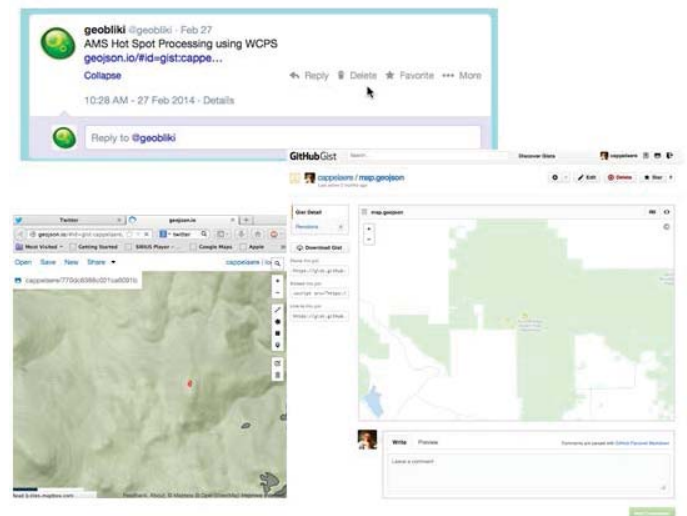


Fig 3 The sample scenario depicts a tweet in the upper part of the figure notifying user of an onboard detection of a fire detection and the subsequent hot pixel product made available via Github and Mapbox

The GeoSocial API is designed to access multiple data sources from multiple disaster nodes. Over time, it is hoped that more servers will adopt the standards needed to interact with the GeoSocial API and thus instead of a centralized catalogue being searched and new products being added to the catalogue, new products publish their availability and then are automatically available.

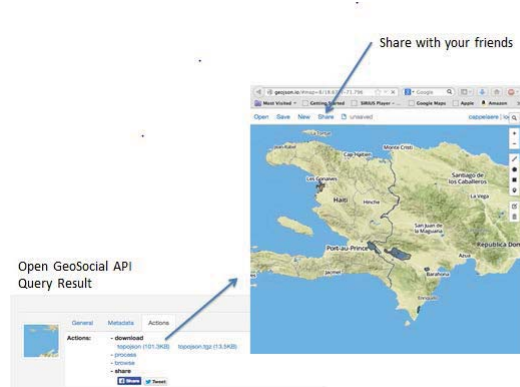


Figure 4 Depicts a different data product from different disaster node

Figure 4 depicts a different product available via the Moderate Resolution Imaging Spectroradiometer (MODIS) product server and then retrieved for viewing or sharing. Note that this is the webserver version of the GeoSocial API. However, the ultimate intent is to enable emergency responders to have access in the field and enable them to upload additional data from the field to augment the satellite data product. Figure 5 shows a preliminary app for the iPhone that performs the key GeoSocial API functions.



Figure 5 Depicts prototype app for the GeoSocial API displaying Earth Observing 1 flood map

III. UNDERLYING SOFTWARE

The underlying software for this architecture leverages existing open software and standards and integrates that capability with previously develop SensorWeb capability from satellites such as Earth Observing 1 that have SensorWeb capability. However, this does not preclude other satellite systems from making use of the GeoSocial API. Adapters would be needed to translate the existing data product chains into formats compatible with the GeoSocial API.

- a. Open Geospatial Consortium (OGC) SensorWeb Enablement (SWE) standards
- b. OpenStreetMap standards
- c. TopoJSON vectorization software
- d. Geojson.io
- e. PostGRES database
- f. PostGIS - open source software program that adds support for geographic objects to the PostgreSQL object-relational database.
- g. Mapbox or Leaflet OSM viewer
- h. Github
- i. Facebook, Twitter
- j. SensorWeb components such as GeoBPMS, GeoBliki and Web Coverage Processing Service (WCPS)

IV. CONCLUSION

The prototyping and capacity building effort on the Namibian Early Flood Warning system which consists of SensorWeb components and Cloud Computing services, support the goals of the CEOS Working Group on Information Systems and Services (WGISS) work. It especially addresses some of the aspects for developing an international disaster architecture which is the focus of some of the work in the WGISS group.

REFERENCES

- [1] Mandl, Daniel, Stuart Frye, Pat Cappelaere, Matthew Handy, Fritz Policelli, McCloud Katjizeu, Guido Van Langenhove, Guy Aube, Jean-Francois Saulnier, Rob Sohlberg, Julie Silva, Natalia Kussul, Sergii Skakun, Stephen Ungar, Robert Grossman, Joerg Szarzynski, "Use of the Earth Observing One (EO-1) Satellite for the Namibia SensorWeb Flood Early Warning Pilot" IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol 6, No. 2, April 2013
- [2] Frye, Stuart, Karen Moe, Dan Mandl, Matt Handy, George Percivall, and John Evans, 2013. "Towards a Sensor Web Architecture for Disaster Management: Insights from the Namibia Flood Pilot", IGARSS 2013, Melbourne, Australia, July 21-26, 2013.
- [3] Mandl, D., "Matsu: An Elastic Cloud Connected to a SensorWeb for Disaster Response", GSAW, Session 12F Working Group: Cloud Computing for Spacecraft Operations March 2, 2011 Los Angeles, CA
- [4] Mandl, D., Rob Sohlberg, Chris Justice, Stephen Ungar, Troy Ames, Stuart Frye, Steve Chien, Pat Cappelaere, Daniel Tran, Linda Derezinski, Granville Pauls, "A Space-based Sensor Web For Disaster Management", IGARSS 2008, July 6, 2008, Boston, MA
- [5] Evans, John, and Karen Moe, 2013. "GEOSS Architecture for the use of Remote Sensing Products in Disaster Management and Risk Assessment". Committee on Earth Observation Systems (CEOS) Working Group on Information Systems and Services (WGISS) report. <http://tinyurl.com/GA4Disasters>

Some of the underlying software and standards includes: